10/530160
Rec'd PCT/PTO 22 SEP 2005
PCT/US2003/020414

RECONFIGURABLE COMPUTE ENGINE INTERCONNECT FABRIC

FIELD OF INVENTION

This invention relates to the interconnection of reconfigurable compute engines and more particularly to an interconnect fabric which itself is reconfigurable to support the different modes of operation for the interconnected reconfigurable compute engines.

BACKGROUND OF THE INVENTION

Reconfigurable compute engines in the past have been connected together so as to provide a computational system which can address many different types of tasks utilizing the same computer elements. Reconfigurable compute engines in general are exemplified by the field programmable gate arrays which when interconnected and appropriately programmed provide such functions as communications, signal intelligence, spatial processing, and jamming. When a suite of electronics is provided in an aircraft, it is oftentimes desirable to be able to reutilize the compute engines by changing their modes on the fly so that the avionics suite performs the above noted functions.

The reason for the utilization of field programmable gate arrays is so that the amount of equipment in an aircraft payload can be minimized due to space and power limitations. However, present field programmable gate arrays, FPGAs, are at present only programmable by programming the entire unit, and thus regardless of the different applications instantiated in the FPGAs oftentimes all interconnects must be changed to effectuate each mode change.

Inherent in the programming of such FPGAs is the ability to configure a physical layer which is coupled to the application programmed into the FPGA so as to provide input

and output pins which carry data, timing, and other signals between the FPGAs that are connected together to provide a particular function.

Thus, it is important that the interconnect between the FPGAs support, inter alia, low deterministic latencies on the order of 1 to 10 nanoseconds and support edge sensitive signal distribution, for instance, for clocks, strobes, triggers and the like. There is also a need for the physical layer to support synchronized data transport.

In general, in order to provide the programming for a system composed of a number of FPGAs, it usually takes 400 man-hours to generate an FPGA configuration, which is preceded by research and development typically on the order of another 400 man-hours. Thus, for each mode of operation of the FPGAs, upwards of 800 man-hours are required.

Were one wishing to change the mode of operation of the FPGAs in an electronics bay of an aircraft one would not want to completely reprogram each one of the FPGAs for each mode of operation, nor rewire the interconnects between the modules. Thus when changing the mode from communications to, for instance, spatial processing, since there is a suite of software which is capable of supporting both applications without alteration, it would be highly desirable to be able to provide a reconfigurable fabric which could selectively connect each of the FPGA modules together and support, for instance, all of the timing required, including strobing, clock triggers and the like, and to, for instance, simultaneously accommodate packet switched networks as well as circuit switched networks. Moreover, whatever interconnect fabric is supplied, it must be as flexible and scalable as possible so as to utilize the various resources and switch between them on demand.

There exist systems where the functionality resides in either general purpose processors, GPPs, or reconfigurable compute engines, RCE, or combinations thereof. Such systems require interconnect fabrics that support a wide range of data flow. While general

purpose processors have interconnect requirements that are being satisfied by current technologies such as Ethernet and Myrinnet, there is no suitable fabric with characteristics appropriate for use between reconfigurable compute engines.

Nowhere is such an interconnect fabric more urgently needed than in command, control, communications and computers used for intelligence, surveillance and reconnaissance. In some initiatives one needs to have a single system that can support communications, signal intelligence, jamming and run on the same hardware. As an example, it is an important requirement to have the hardware configured to have a number of receive channels and a number of transmit channels and to reconfigure the RCEs either to perform a communications function, signal intelligence or jamming.

As mentioned above, the RCEs presently are realized using FPGAs, with the combination of the FPGAs, for instance, in communications having a tuner that feeds them, with the FPGAs then feeding some processor and feeding back up to another FPGA and then to a transmitter. The interconnection protocols for connecting the FPGAs together provide for the receive and the transmit functions associated with communications.

On the other hand signal intelligence systems have only a tuner, FPGA and a processor. It will be appreciated that when operated in this mode there is no transmission so that only one half of the system normally configured for communications need be used. A jammer usually has processing followed by a FPGA, with the output of the FPGA then coupled to a transmitter so that the jammer consists in essence of a receiver and processing, with the output utilized to drive a transmitter for counter measure purposes.

In any avionics suite, there may be up to a dozen FPGAs which are connected to do many of a variety of functions. For instance at any given time, one might utilize five FPGAs to do communications and two to do signal intelligence. Later one might want to change the

avionics around so that other FPGAs are connected together to provide, for instance, a spatial processing task or a jamming function.

In the past, one would utilize a well-defined static interconnection system for whatever mode or function was required to enable one FPGA to talk with another one.

However, because multi-mission systems are desired, it is important to be able to make the avionics package completely reconfigurable so that any FPGA can communicate with any other FPGA. In order to do so one needs a data fabric to allow the FPGAs to exchange data at high rates with low and deterministic latencies.

In order to connect the FPGAs, at the physical layer, one is talking about the ability to sustain a certain bandwidth on a given twisted pair. For twisted pairs one considers point-to-point terminations and the twisted pairs should be able to accommodate an edge sensitive strobe. It should be able to accommodate packet switched data, circuit switched data, or to accommodate fixed point-to-point connections with synchronized or asynchronized data.

Beyond the physical layer, one needs to be able to provide for different protocol stacks. Depending on the function implemented in any one of the reconfigurable compute engines, the particular protocol stack determines the kind of signaling whether it be discrete strobes, packet switched or circuit switched type of topologies or the like which have to be supported.

In short, one needs to have a common suite of hardware that would perform the multi-mission or multi-mode architecture. One therefore needs a fabric that has the ability to support all the different interconnect methods without having to go and change the underlying hardware.

Not only must the interconnect fabric support all kinds of clocks, it must also support those whose clock rates require that there be some minimum number of clock edges. It must

also accommodate 8B/10B encoded signals and for instance, must support 10 MHz square wave clock pulses. The fabric must also be able to support discrete signals or signals which involve a simple level which is changed.

While the above lists some of the types of signals that one could expect, one also needs to have the fabric support a number of different mode changes without having to reload the FPGA. In short, one could think of the fabric that needs to be provided as a distributed switch architecture where the intelligence lies in each reconfigurable compute node.

In short, the fabric or physical structure should be able to support two modes. One mode is one in which there are parameters that are set as bits in a register or are embedded in each reconfigurable compute engine that defines what pins are to exist at one FPGA node. The pins are arranged such that they are either source pins, destination pins or pass through pins. However, just supporting the above while supporting a limited number of reconfiguration states is insufficient. What is required is a structure that does not lend itself to just being a bit in a register. What is necessary is the reloading of some number of the FPGAs so that complimentary configurations can be instantiated so that several of the FPGAs can talk to each other.

In short, what is necessary is logic that is installed in the FPGAs to instruct the way that the FPGAs are interconnected and allows one to change the way they are interconnected without having to reload the FPGA or without having to rewire them. What is therefore necessary is to have some modality to a virtual circuits set up that can be torn down at any time that a mode change needs to be made. With the respect to the pins for the FPGAs, a pin on a node needs to be able to be configured to be a transmitter, a receiver or beyond that to function as a pair for differential signal arrangements that can accommodate edge sensitive reconfigurable strobes or encoded signal structures. A reconfigurable fabric also needs to be

able to accommodate either packet switched topologies or circuit switched topologies or combinations of each, as well as accommodating singleton kinds of implementations. Moreover, the fabric needs to be able to accommodate optical translations.

SUMMARY OF THE INVENTION

In order to accommodate all of the above functions, in the subject inventions an interconnect mechanism is provided that can be dynamically reconfigured to support scalable connection-oriented packet switched networks and connection-oriented circuit switched interconnects between multiple reconfigurable compute engines. The connections need to be made all at the same time so that there is no sharing and for all intents and purposes looks like a cross point array. The subject under interconnect mechanism or fabric supports multi-cast, broadcast, or single node-to-node embodiments.

It should be noted that the difference between the fabric and a bus is that with a bus sequential events are made to occur, where in a fabric simultaneous signaling occurs. In one embodiment, assuming that the RCE is an FPGA, each of the FPGAs is provided with an applications program, an interconnection layer, and a physical layer. It is the purpose of the interconnection layer to be reconfigurable regardless of what is happening with the application within the FPGA and to set the connections for the physical layer both as to what connections corresponds to what operation, and also the modality of the pin itself, whether it be a receiver pin, a transmitter pin, or a timing connection.

Thus, rather than having to completely reload a field programmable gate array, in the subject invention all that is necessary to interconnect the field programmable gate arrays for a given mode or function is to reconfigure the interconnection layer. This reduces programming from the aforementioned 800 man-hours to as little as 200 man-hours.

The subject invention thus provides the user with the flexibility of dynamically defining the modes and the interconnects necessary between the RCEs.

By not having to reconfigure each and every FPGA for a mode change, one can utilize a standardized physical layer which does not have to be changed during mode changes and which does not require a complete reload of the FPGA.

While companies that manufacture FPGAs design the input and output pins so that they can physically be inputs or outputs, and then can be driven at different voltage levels, utilization of the subject programmable interconnect layer brings the programming to another level. The interconnect layer defines the functions of the pins and the functions of groups of pins as well as describing or designating the data and where it is to flow.

In short, the interconnect layer provides an interface to the physical layer from the application by providing a variety of physical mechanisms and switched topologies so that the application is abstracted from the physical interconnect. In one embodiment, the system allows processing components to communicate over one or more FPGAs without specific knowledge of the interconnect structure and without components having specific knowledge of the particular topology, particular cards or anything else. This is because the application is abstracted by the interconnect layer which actually handles setting up the particular routing required, essentially at instantiation time. In short, the system decouples the application components from needing to know anything about the particular fabric, meaning that the subject system provides a very rich interconnect service.

Thus when a core or cores are loaded into an FPGA, the FPGA interconnect IP core is positioned between the application or applications instantiated in the FPGA and the physical layer supported by a given vendor. Note that the interconnect layers are that which are

distributed amongst the FPGAs, with complementary parts of the interconnect layer being provided at the different FPGAs.

When an aviation suite is provided in such a manner, one can in one instance do spatial processing which requires a specific data flow to be supported. This would include the classic scatter gathering problem across a number of input sensors and destinations. For communication applications, one has some number of tuners and transmitters that are supporting different signal structures from different radios. Thus the subject fabric has to bridge data from one receiver to a transmitter and this may occur multiple times.

For instance, in order to do a mode change to support signal intelligence one abandons the communications support and may double the amount of frequency spectrum that one can cover by allocating all of the receive channels to support nothing but spatial processing and by implementing the scatter gathering problem across all of the tasked nodes. It will be appreciated that in so doing no new wiring of any kind is required in subject fabric. Additionally, the subject fabric is able to support the low and deterministic latency required.

If for instance one wants to go from communications to signal intelligence one may need to allocate all of the resources to this problem and to do so without rewiring or writing different software.

The utilization of the applications unaltered by the interconnect also and importantly eliminates the problem of providing a large amount of non-recurring engineering.

With the above reconfiguration management, all of the equipment is reusable providing increased utilization and efficiency. One obviously first writes something that is generic and reprogrammable from an interface point of view and provides one wrapper that helps plug in the standard interconnect logic that never has to be changed. One can change topologies, functions, and change mixes of functions, and do so only by throwing the

switches in the subject interconnect layer provided between the application and the physical layer. Thus, even if one has ten different topologies or configurations, one does not have to redo the interconnect layer of the FPGAs ten times. As a result what is accomplished is efficient mode change both from the point of view of design and from the point of view of implantation.

As will be appreciated, the subject fabric combines the ability to support deterministic, low latency interconnects versus simple packet switching which generally does not have deterministic latency. As will be appreciated latency does not always end up being as short as one would like it to be. With the subject fabric one can accommodate packet switched topologies or circuit switched topologies and quite often applications that require both. One can have messaging to get from one node to another which is not necessarily time critical; but there are applications where accompanying strobes are absolutely critical.

In summary, a reconfigurable compute engine interconnect fabric includes a reconfigurable interconnect layer within each reconfigurable compute engine between the application loaded into the engine and the physical layer used for inputs to and outputs from the engine. The reconfigurable interconnect layer identifies the input and output pins for the engine and their functions. Thus, a system made up of reconfigurable compute engines can be changed by loading appropriate applications into respective engines and connecting complementary engines by defining the pins and specifying their functions using the reconfigurable interconnect layer such that no physical connecting structure need be changed to effectuate a mode change. The fabric supports strobes, clocks and various forms of data transfer including circuit switched, packet switched and discrete signal modes. Moreover, the fabric is flexible enough to support deterministic low latency interconnects or simple packet switching which does not have deterministic latency, node to node messaging and both

packet switched and circuit switched topologies. In one embodiment, the reconfigurable compute engines include FPGAs.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the subject invention will be better understood in connection with the Detailed Description in conjunction with the Drawings, of which:

Figure 1 is a diagrammatic representation of the utilization of reconfigurable compute engines interconnected variously to provide spatial processing, communications, signal intelligence and jamming in which different ones of the reconfigurable compute engines are connected together through the subject interconnect structure or fabric;

Figure 2 is a diagrammatic representation of the utilization of an interconnect layer between an application layer and a physical layer in complementary FPGAs;

Figure 3 is a diagrammatic representation of the utilization of a number of different reconfigurable compute engines coupled together by their physical layers so as to support strobing, the data flow, discrete signaling through the utilization of a reconfigurable interconnect mechanism, the mechanism being the reconfigurable layer of Figure 2;

Figure 4 is diagrammatic illustration of the utilization of a bus structure to interconnect general purpose computers, illustrating the sequential nature of the bus;

Figure 5 is a diagrammatic illustration of a cross point structure which would, for instance, connect transmit nodes of various RCEs to receive nodes utilizing the subject interconnect fabric;

Figure 6 is a block diagram of the interconnection of a number of reconfigurable compute engines, each with a different function, interconnected with the subject fabric, with

general purpose processors connected together with their own fabric, and with the reconfigurable compute engines coupled to respective general purpose processors; and,

Figure 7 is a block diagram illustrating the utilization of an interconnect fabric for the interconnection of several reconfigurable compute engines, with the first RCE having pins defining data transmitters, an unused pin and a defining a strobe receiver, and with a complementary reconfigurable compute engine having corresponding data receivers, an unused pin and a strobe transmitter, with the data transmitter/receiver connection representing a connection oriented circuit switched data path, with the unused connection representing an unused physical path, and with the strobe transmitter/receiver connection representing a connection oriented circuit switched event strobe path.

DETAILED DESCRIPTION

Referring now to Figure 1, in an avionics suite a number of reconfigurable compute engines 10 may be interconnected so as to perform a number of functions. One of the functions may be a spatial processing function illustrated at 12 in which various of the physical layers of the reconfigurable compute engines are interconnected to provide a spatial processing function. Assuming a communications function 14 is desired, then selected reconfigurable compute engines are connected together for the communications function. Likewise for a signal intelligence function 16, various of the reconfigurable compute engines are interconnected as illustrated, whereas for a jamming function 18 the reconfigurable compute engines may be arranged, for instance, in a ring topography and connected by the subject fabric. As illustrated by dotted line 11, in the ring topography scheme direct connection of RCEs is possible.

In order to provide for the reconfigurable interconnect structure shown in Figure 1, each of the reconfigurable compute engines as illustrated in Figure 2 may include an FPGA 20 with each of the FPGAs having an applications layer 22 an interconnect layer 24 and a physical layer 26 that interconnects the two FPGAs.

It is the purpose of this invention to be able to interconnect complementary FPGAs through the use of an invariant physical layer in which the interconnect layer defines the input and output for each of the FPGAs both as to the type of pin that they represent and also which pins are connected to what pins between the FPGAs.

With this arrangement, the applications loaded into the FPGAs may be made to change depending on the mode of operation for the entire system, with the physical layer being a fabric which connects the pins of complementary FPGAs together under the control of the interconnect mechanism which resides in the FPGA interconnect layer.

Referring to Figure 3, each of the RCEs 30, 32, 34 and 36 has a physical layer 40 which constitutes a reconfigurable interconnect mechanism under the control of the interconnect layer within the RCE. As can be seen, one interconnect between RCE 30 and RCE 36 supports a strobe 42, whereas for instance, the interconnect 44 between RCE 34 and 36 supports data flow between these two RCEs, with the data flow, for instance, being either circuit switched or packet switched. Other types of data can be transmitted between, for instance, RCE 36 and RCE 32 over an interconnection 46 which supports, for instance, discrete signaling in terms of the level of a signal transmitted.

The subject system is different from the bus structure shown at 50 in which a number of general purpose processors 52 are interconnected by a bus 54 to which are applied time sequenced or multiplex signals such that the interconnection between the general purpose processors is not done in parallel but rather is done serially.

Referring to Figure 5, what is shown in general is a cross point connection matrix 60 diagrammatically illustrating how complementary FPGAs can be connected by virtue of a fabric. Here a number of transmitter pins 62, 64, 66 and 68 are selectively connected to complementary receiver pins 70, 72, 74 and 76, with the cross point fabric providing the connection of transmitter pin 66 to receiver pin 74 overlying 80 whereas transmitter pin 64 is connected to receiver pin 76 over line 82, with transmitter pin 62 connected to receiver pin 72 over line 84. What will be appreciated is that the connections provide a parallel connection as opposed to the serial connection associated with any type of bus structure.

As can be seen from Figure 6, a number of RCEs 90, 92, 94 and 96 are provided respectively with various functions 98, 100, 102 and 104. These are the functions that are required to be performed by of the RCEs to effectuate a given mode of operation. The subject fabric 110 is utilized to interconnect all of the RCEs.

As can be seen, each of the RCEs is connected to its associated general purpose processor, here respectively illustrated at 112, 114, 116 and 118 interconnected with a second fabric 120 of a conventional Ethernet or other variety.

The entire system illustrated in Figure 6 can made to execute a number of functions depending on the functions f1-f4 loaded into the respective RCEs. Fabric 110 interconnects the RCEs in such a manner as to effectuate the particular mode.

Referring to Figure 7, in one example, a number of RCEs 130, 132 and 134 are interconnected by connecting various pins 136, 138, 140 and 142 on RCE 130 to complementary pins 136', 138', 140' and 142'.

Thus for instance, data transmitter pins 136 and 138 on RCE 130 are connected to complementary pins 136' and 138' respectively for RCE 132, which pins correspond to data

receiver pins. The connections designated by arrows 144 correspond to a connection oriented switched circuit data path between RCE 130 and RCE 132.

Likewise a path 146 between strobe transmitter pin 142' at RCE 132 is established by connection to a strobe receiver pin 142 on RCE 130. Note that path 146 represents a connection oriented circuit event strobe path in this case.

What will be appreciated is that the interconnect layer for each of the RCEs defines the pin and what the function of the pins is so that the physical layer remains invariable, with the pins and their functions being determined by the interconnect layer. Thus the physical layer remains unchanged, whereas the interconnect functions are programmable through the programming of the interconnect layer. Note that the programming of the interconnect layer does not affect either the application or the physical layers so that mode changes can be rapidly effectuated utilizing the same or lesser numbers of available RCEs.

Having now described a few embodiments of the invention, and some modifications and variations thereto, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by the way of example only. Numerous modifications and other embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention as limited only by the appended claims and equivalents thereto.